ARNOLD: A Benchmark for Language-Grounded Task Learning With Continuous States in Realistic 3D Scenes

Ran Gong[1]*, Jiangyong Huang[2,5]*, Yizhou Zhao[1], Haoran Geng[2,5], Xiaofeng Gao[1], Qingyang Wu[4], Wensi Ai[1], Ziheng Zhou[1], Demetri Terzopoulos[1], Song-Chun Zhu[2,3,5], Baoxiong Jia[5], Siyuan Huang[5]

[1]University of California, Los Angeles, [2]Peking University, [3]Tsinghua University, [4]Columbia University, [5]National Key Laboratory of General Artificial Intelligence, BIGAI

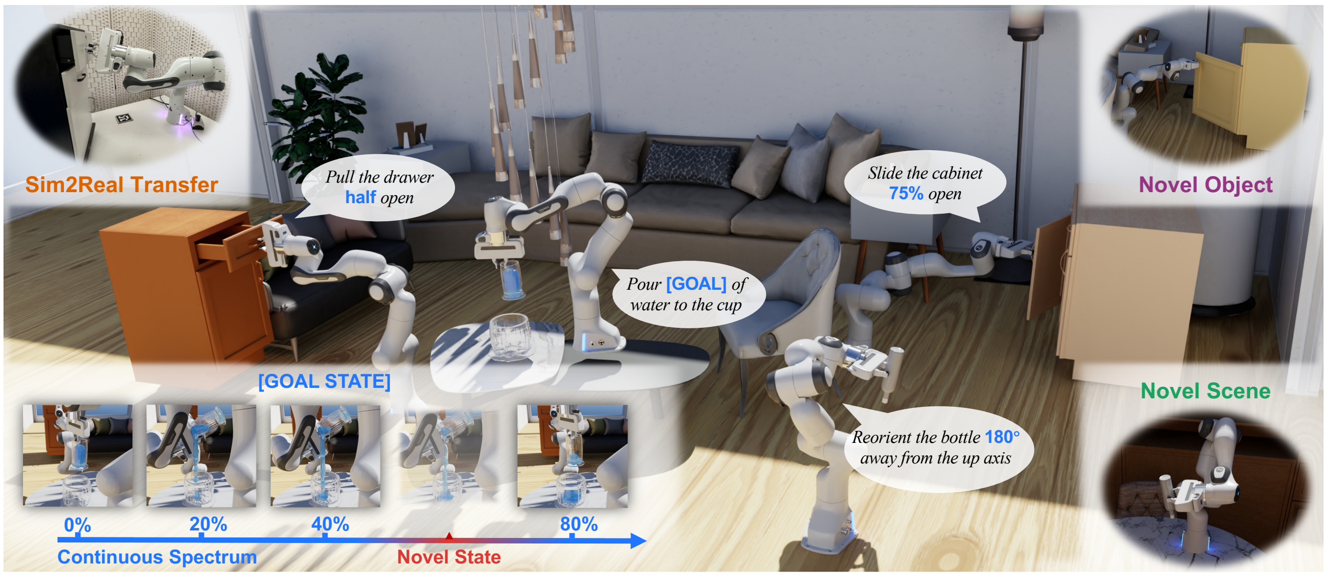

Figure 1. **The ARNOLD benchmark** for language-grounded task learning with continuous states in realistic 3D scenes. ARNOLD provides 8 tasks with their demonstrations for learning and a testbed for the generalization abilities of agents over (1) novel goal states, (2) novel objects, and (3) novel scenes.

Abstract

Understanding the continuous states of objects is essential for task learning and planning in the real world. However, most existing task learning benchmarks assume discrete (e.g., binary) object goal states, which poses challenges for the learning of complex tasks and transferring learned policy from simulated environments to the real world. Furthermore, state discretization limits a robot's ability to follow human instructions based on the grounding of actions and states. To tackle these challenges, we present ARNOLD, a benchmark that evaluates language-grounded task learning with continuous states in realistic 3D scenes. ARNOLD is comprised of 8 language-conditioned tasks that involve understanding object states and learning policies for continuous goals. To promote language-instructed learning, we provide expert demonstrations with template-generated language descriptions. We assess task performance by utilizing the latest language-conditioned policy learning models. Our results indicate that current models for language-conditioned manipulations continue to experience significant challenges in novel goal-state generalizations, scene generalizations, and object generalizations. These findings highlight the need to develop new algorithms that address this gap and underscore the potential for further research in this area.

Project website: https://arnold-benchmark.github.io.

1. Introduction

The ability to ground language is a crucial skill that has evolved over the course of human history, allowing people to learn and describe concepts, perform tasks, and communicate with one another. While recent developments have

* Equal contribution. Corresponding authors: Ran Gong, Jiangyong Huang, Baoxiong Jia, and Siyuan Huang.

enabled the grounding of concepts in images [66, 39, 68, 21], the interaction with physical environments [10, 2, 83, 43, 61, 36, 35, 28, 26, 24, 86, 78, 9, 15], and language understanding in physical environments [49, 1, 64, 88, 76, 22, 79, 63], few researchers have investigated the grounding of actions in daily tasks [70, 89, 71, 80]. Given that humans can comprehend object status and relate language instructions to the physical environment, a pertinent question arises: *How can we imbue robotic systems with the same capacity to understand and execute language instructions in the physical world?*

Learning action grounding in daily activities is a challenging task that presents several non-trivial difficulties. Firstly, robotic tasks rely heavily on detailed scene understanding for successful execution. This includes the understanding of geometry information, layouts, and visual appearances. The various combinations of scene configurations, including novel appearances, objects, and spatial positions further exacerbate this challenge. Therefore, it is crucial for robotic systems to acquire generalizable skills that can be transferred to different domains and settings.

Secondly, humans possess an exquisite ability to understand desired goal states precisely. This ability allows us to effortlessly map from simple descriptions (*e.g.*, a cup half filled, a door fully opened, *etc.*) to the precise status of physical properties (*e.g.*, half the volume, pulled to 180°, *etc.*). However, it is exceedingly challenging for robots to learn the precise goal state from abstracted language instructions, especially when referring to an implicit range of continuous object states (*e.g.*, a bit of coffee, slightly open, *etc.*) [41, 30, 56]. As a result, there is an urgent need for robot systems to maintain a mapping from language instructions to precise goal states in a continuous world.

A necessary first step toward tackling these challenges is to develop realistic robot simulation systems that enable language-grounded learning. Indeed, notable recent advances in simulated environments have facilitated grounded task learning [8, 72, 57, 89, 53]. Despite the impressive progress, these benchmarks suffer from several limitations that hinder the effective operation of robots in the real world: (1) They typically assume that tasks are performed in simple and clean environments, rather than in scenes that are spatially occupied by clutter and visually disturbed by diverse textured backgrounds [42, 45, 89, 71]. (2) They assume discrete (*e.g.*, binary) object states and perfect motor control that ignore the low-level geometry and dynamics of objects [75, 73, 16] and, consequently, they do not attempt in-depth physical state understanding or fine-grained manipulation skills. (3) These benchmarks do not ground instructions to precise states [89, 70], thus neglecting the challenging problem of grounding language to object states.

To address these critical challenges of language-grounded robot task learning, we introduce a new benchmark, ARNOLD, for grounding task instructions to precise robot actions and object states in realistic natural scenes (Fig. 1). Specifically, we leverage a highly accurate physics simulation engine to create eight challenging robot manipulation tasks that include continuous robot motion, friction-based grasping, and a variety of object state manipulations. Each task is associated with a set of goals sampled from a continuous range of object states and their corresponding detailed task descriptions in human language form. We further provide plentiful demonstrations of each task with trajectories generated with a template-based planner for robot learning.

To provide an in-depth evaluation of language-grounded task learning, we complement prior research with an evaluation that targets the ability of agents to generalize learned language-grounded skills to unseen scenarios, including novel scenes, novel objects, and our featured novel goal states. We have meticulously curated a collection of 40 distinctive objects and 20 scenes from open-source datasets [84, 40, 18] and designed data splits for evaluating different aspects of agents' generalization ability in language-grounded task learning. Furthermore, we provide thorough experimental analyses and show that state-of-the-art language-conditioned manipulation models still suffer with regard both to grounding and generalization. Additionally, we show that state modeling is crucial for tasks in ARNOLD through carefully designed ablation studies.

In summary, ARNOLD makes the following contributions:

- A **realistic 3D interactive environment** with diverse scenes, objects, and **continuous object states**, facilitating the learning and evaluation of precise robot manipulation.
- A systematic benchmark comprising eight challenging **language-grounded robotic tasks** and evaluation splits for different aspects of **skill generalization**.
- **Extensive experiments and analyses** of state-of-the-art language-conditioned manipulation models, revealing their strengths and weaknesses in promoting future research on language-grounded task learning.

2. Related Work

Simulators for Embodied AI. Significant progress has recently been made in developing simulators for training and evaluating AI agents to perform indoor household activities [43, 17, 19, 6, 11, 5]. To mitigate complexity, most of these simulators make simplifications about world states and actions, abstracting robot manipulation into symbolic planning in discrete action [65, 40] and state spaces. However, agents trained in such settings are unaware of the relationship between actions and the geometries and dynamics of objects, therefore limiting their abilities in real-world scenarios. Recent efforts have gradually transitioned to continuous action spaces, but they still make some simplifications. For example, grasping is often simplified by attaching a nearby object to the gripper [16, 73], or through contact [44, 37, 75].

Table 1. **Comparison with existing benchmarks.** ARNOLD features language-grounded robot control over continuous object states with a large number of demonstrations in photo-realistic scenes. ARNOLD also **leverages advanced physics simulations powered by PhysX 5.0 to simulate articulated bodies and fluids**. **Language:** Task goals are specified by human language instruction. **Multi-Camera:** Robot is equipped with multiple cameras. **Fluid:** Advanced fluid simulation. **Physics:** Realistic physics simulation with realistic grasping. [1]: RLbench-based benchmarks use simplified grasping. **Continuous:** Object state and goal state are continuous. **Scene:** Tasks are performed with a realistic scene background. **Robot:** Perform actions with real robots for all tasks. **R**: Rasterization. **RT**: RayTracing. **Flexible Material**: Easy to change materials and textures. **Generalization**: Systematic generalization test at different levels.

Benchmark	Language	Multi Camera	Fluid	Physics	Continuous	Scenes	Robot	Rendering	Flexible Material	Generalization
Alfred [72]	✓	✗	✗	✗	✗	✓	✗	R	✗	✗
Maniskill [59, 29]	✗	✓	✓	✓	✓	✗	✓	R	✗	✓
Calvin [57]	✓	✓	✗	✓	✗	✓	✓	R	✗	✓
Behavior [44, 73]	✗	✗	✓	✓	✓	✓	✓	RT	✓	✗
KitchenShift [85]	✗	✗	✗	✓	✗	✗	✓	R	✗	✓
RLBench [37]	✓	✓	✗	✓[1]	✗	✗	✓	R	✗	✗
Softgym [45]	✗	✗	✓	✓	✓	✗	✗	R	✗	✗
Orbit [58]	✗	✗	✓	✓	✗	✓	✓	RT	✓	✗
Vlmbench [89]	✓	✓	✗	✓[1]	✓	✗	✓	R	✗	✗
Ravens [87, 70]	✓	✓	✗	✓	✗	✗	✓	R	✗	✓
Habitat HAB [75]	✗	✗	✗	✗	✗	✓	✓	R	✗	✓
TDW Transport [19, 20]	✗	✗	✗	✗	✗	✓	✓	R	✗	✗
ARNOLD	✓	✓	✓	✓	✓	✓	✓	RT	✓	✓

Among works that provide continuous object state change simulation, most do not focus on manipulating object states in a precise and fine-grained manner. For example, VRKitchen [23] defines task goals in a discrete manner even though the underlying object states are continuous. Softgym [45] is an object manipulation benchmark that provides a realistic simulation of deformable objects; however, it lacks diversity among objects and scenes.

By contrast, ARNOLD provides a wide variety of scenes and objects. And ARNOLD simulates continuous states for articulated objects and simulates fluids at the particle level. We control the robots with 7-DOF continuous control and friction-based grasping powered by a state-of-the-art physics engine (PhysX 5.0). Whereas most of the environments [29, 75] optimize for speed, we optimize for the realism of the rendering. And ARNOLD also equips a remarkable rendering speed at 185 FPS (37 FPS with five cameras).

Language Conditioned Manipulation. Relating human language to robot actions has been of recent interest [50, 74, 89, 38, 12, 33, 32, 62]. However, the environments in these efforts either lack realistic physics [72, 70] or do not have realistic scenes [71, 57] where the surroundings of the agent will constrain its motion, and different scene objects might occlude the agent's viewpoint. Additionally, systems like [33, 4, 14, 51] are application-based, lacking a systematic benchmark for language-conditioned manipulation. Most importantly, prior work aims to ground human language to static object properties, such as colors and shapes. By contrast, ARNOLD provides instructions for continuous object states. We compare between ARNOLD and other related benchmarks in Tab. 1.

Continuous State Understanding. Some recent research tries to predict object states [47, 60]. However, the object states are discrete rather than continuous. More recently, researchers [82, 13, 81, 77] tried to predict object states continuously. However, they do not address the manipulation of objects from arbitrary starting states to the desired states. Moreover, they do not model the language grounding process. Most recently, Ma *et al.* [52] propose a method to perform precise object state manipulations, but their approach does not perform language grounding and only a small subset of our tasks are covered with their simple motion primitives.

Compared with prior works, we provide more diverse goal states to cover the continuous state space instead of learning only binary goal states. This allows models to understand the continuous state space. On the other hand, we also propose evaluation of generalization in terms of continuous state understanding (see Sec. 3.4). This evaluates how the model leverages its understanding of the state space to generalize within a continuous spectrum, which is rarely studied in prior works. Though more goal states can be added with our continuous simulation, we leave ARNOLD at the current scale since more states will lead to higher costs of data generation due to the compositions of object/scene/state.

3. The ARNOLD Benchmark

The ARNOLD benchmark is motivated by the abilities that an intelligent manipulator agent should possess, including (1) the ability to comprehend and ground human instructions to precise world states, (2) the capacity to acquire policies for generating accurate actions and plans toward precisely defined goal states, and (3) the feasibility of transferring such abilities to the real world. Therefore, in ARNOLD we focus on language-conditioned manipulation driven by continuous goal states situated in diverse photo-realistic and physically-realistic 3D scenes.

3.1. Simulation Environment

Simulation Platform. ARNOLD is built on NVIDIA's Isaac Sim [55], a robotic simulation application that provides photo-realistic and physically-accurate simulations for robotics research and development. In ARNOLD, the photo-realistic rendering is powered by GPU-enabled ray tracing, and the physics simulation is based on PhysX 5.0. Fig. 1 and Fig. 2 provide examples of simulation and rendering.

Physical Simulation. To ensure physically-realistic simulation, we assign physics parameters to objects, including weight and friction for rigid-body objects, and cohesion, surface tension, and viscosity for fluids. These parameters are selected as in prior work [59] and are adjusted by human operator feedback. Fluids are simulated using the GPU-accelerated position-based-dynamics (PBD) method [54] through NVIDIA's Omniverse platform. Depending on the rendering speed, we perform an optional surface construction process using marching cubes [48] to achieve the final fluid rendering effect.

Scene Configuration. There are 40 distinct objects and 20 diverse scenes in ARNOLD. The scenes are curated from [18], a large-scale synthetic dataset of indoor scenes. This endows ARNOLD with professionally designed layouts and high-quality 3D models. In addition to objects provided by Isaac Sim, we collected objects from open-source datasets [40, 84]. We modified object meshes to enhance visual realism, *e.g.*, by modifying materials and adding top covers to cabinets and drawers. For more stable physics-based grasping, we performed convex decomposition to create precise collision proxies for each object. More details are found in Appendix A of [27].

Robot. We use a 7-DoF Franka Emika Panda manipulator with a parallel gripper in ARNOLD for task execution. The agent has direct control over its seven joints and its gripper. We represent end-effector actions with three spatial coordinates for translation and quaternion for rotation, as it is more tractable for policy learning [46]. We utilize the built-in motion planner of Isaac Sim to transform the end-effector action back to the space of robot joints for execution. Currently, our tasks do not involve navigation, *i.e.*, the robot base remains fixed during task execution.

Visual Input. In ARNOLD, we use five cameras around the robot for visual input. As shown in Fig. 2, the cameras provide various views, including front, left, robot base, and wrist. While each camera provides RGB-D input at a resolution of 128×128 by default, users can render at arbitrary resolution. Notably, unlike the deterministic rendering in prior works [71, 89], the rendering in ARNOLD is stochastic due to the ray tracing sampling process [69], which makes ARNOLD more realistic and challenging. In addition to the visual observation, other auxiliary observations can be accessed, *e.g.*, camera parameters, robot base pose, and part-level semantic mask. Other Omniverse sensors (*e.g.*, tactile) are excluded here since they are not required by the tasks and models. They are all available if necessary.

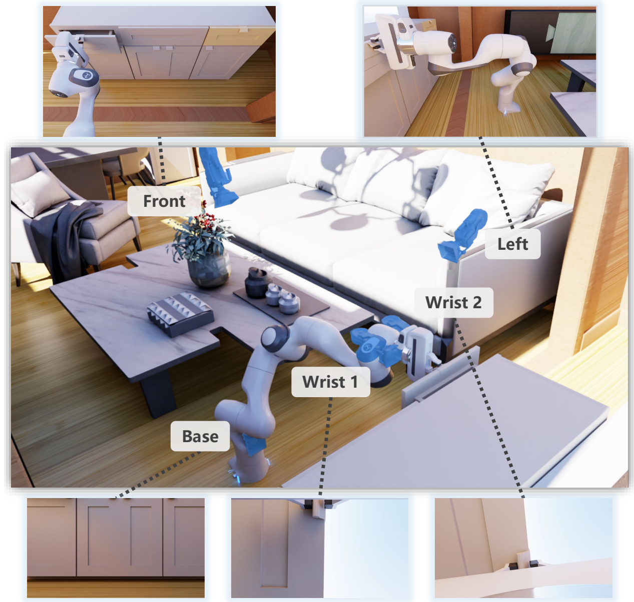

Figure 2. Multi-view robot observation in ARNOLD. The top row shows views from the front and left cameras and the bottom row from the base and two wrist cameras.

3.2. Task Design

We include eight tasks with various goal state variations in ARNOLD. Specifically, we focus on continuous goal states and define success ranges around them wherein robots should maintain object states for 2 seconds to succeed. Tab. 2 provides an overview and Fig. 1 a visualization. More illustrative examples are shown in Appendix B of [27] . Performing these tasks requires capabilities in language grounding, friction-based grasping, continuous state understanding, and robot motion planning. Additional task details follow:

- In PICKUPOBJECT and REORIENTOBJECT, we instruct the robot to manipulate a bottle to achieve different goals. For the former, the initial state of the object is on the ground with goals specifying heights above the ground. For the latter, the initial state of the object is on the ground, oriented horizontally (the state value equals 90°), with goals specifying the angle between the object's orientation and the upright orientation.
- In the four tasks {OPEN,CLOSE}{DRAWER,CABINET}, the goal value specifies the geometric state of the articulated joint, either in terms of distance (for prismatic joints in DRAWER) or angle (for revolute joints in CABINET). The initial state is any value smaller than the goal for OPEN and larger than the goal for CLOSE.

Table 2. Overview of the 8 tasks in ARNOLD. Each task features 4 goal states specified by human language, one of which is reserved for novel state evaluation. The task is deemed successful when the object state remains in the success range for two seconds. Note that TRANSFERWATER imposes the extra condition that only less than 10% of the original amount of water in the cup can be spilled.

Task Types	Goal States	Success Ranges
PICKUPOBJECT	10, 20, 30, 40 (cm)	±5 cm
REORIENTOBJECT	0, 45, 135, 180 (°)	±20°
OPENDRAWER	25, 50, 75, 100 (%)	±10%
CLOSEDRAWER	0, 25, 50, 75 (%)	±10%
OPENCABINET	25, 50, 75, 100 (%)	±10%
CLOSECABINET	0, 25, 50, 75 (%)	±10%
POURWATER	25, 50, 75, 100 (%)	±10%
TRANSFERWATER	20, 40, 60, 80 (%)	±10%

- In POURWATER and TRANSFERWATER, the manipulated object is a cup containing water, and the goal specifies the percentage of water to be poured out (POUR) or poured into another cup (TRANSFER). In these two tasks, the goal values are specified as percentages of water relative to the initial amount of water in the cup.

Our task pool covers a variety of manipulation skills and grounding aspects. PICKUPOBJECT and REORIENTOBJECT are selected for the basic skills of moving and rotating objects and the grounding of distances and angles. These abilities are then composed and reinforced in the four tasks {OPEN,CLOSE}{DRAWER,CABINET}, where the goal state is grounded on the state of the manipulated drawer or cabinet joint. Beyond rigid-body objects, fluid manipulation in the two tasks {POUR,TRANSFER}WATER challenges the robots' ability to manipulate containers and move fluid, grounding goal state values to fluid volumes.

3.3. Data Collection

Demonstration Generation. We designed a motion planner for each task to generate demonstrations. We partitioned each task into sub-task stages for the planner. For each stage, we adopted the RMPflow controller [7] to plan motions toward keypoints. Unlike other approaches to data curation in simulation environments, this keypoint-based motion planner approach affords high sampling efficiency and facilitates imitation learning. While motion planning appeared to be challenging on particular tasks, as demonstrated in [59, 29], we introduced some prior design and practical techniques (details in Appendix B of [27]) to produce satisfactory outcomes. For example, we leveraged spherical linear interpolation (Slerp) to accommodate continuous manipulation in the CABINET and WATER tasks. As a result, our motion planner can efficiently generate demonstrations.

Augmentation With Human Annotations. Despite the strength of motion planners, the diversity of produced demonstrations is highly dependent on the keypoints. To mitigate this problem, we collected about 2k human annotations of task configurations (*e.g.*, object positions), which amount to considerably more diverse and higher quality data. Moreover, we augmented the data with additional relative positions and robot shifts to broaden data variations. Eventually, we curated demonstrations by running inference with ground-truth keypoints and verifying the validity of initial configurations in each execution. In total, we collected 10k valid demonstrations for the ARNOLD benchmark (as in Tab. 3), with each demonstration containing 4–6 keyframes.

Table 3. Dataset statistics. (1) *Train*: training data. (2) *Val*: validation data for model selection. (3) *Test*: test data for i.i.d. evaluation. (4) *Object/Scene/State*: *Novel* splits for generalization evaluation. (5) *State**: the *Any State* split for generalization on arbitrary state.

	Train	*Val*	*Test*	*Object*	*Scene*	*State*	*State**	Total
PICKUPOBJECT	623	134	134	275	221	294	134	1,815
REORIENTOBJECT	355	76	77	114	82	210	77	991
OPENDRAWER	554	119	119	155	255	348	119	1,669
CLOSEDRAWER	671	147	148	251	81	530	148	1,976
OPENCABINET	319	69	69	81	181	241	69	1,029
CLOSECABINET	478	103	103	55	157	72	103	1,071
POURWATER	312	67	67	96	87	186	67	882
TRANSFERWATER	259	56	56	51	50	119	56	647
Total	3,571	771	773	1,078	1,114	2,000	773	10,080

Language Instructions. For each demonstration, we sampled a template-based language instruction with our language generation engine. We designed several instruction templates with blanks for each task, and each template can be lexicalized with various phrase candidates. For example, the template "*pull the* [position] [object] [percentage] *open*" may be lexicalized into "*pull the top drawer* 50% *open*". In addition to the representation by explicit numbers, we also prepared a candidate pool of equivalent phrases (*e.g.*, "*fifty percent*", "*half*", "*two quarters*") for random replacement. Note that the instruction does not specify the initial state, so the agent must recognize the current state from the observation. We present template examples in Appendix C of [27].

3.4. Benchmark

Data Split. Evaluating and improving the generalization abilities of robots is a major focus of ARNOLD. To this end, we randomly split the objects, scenes, and goal states into seen and unseen subsets, respectively. We then created the *Normal* split by gathering data with seen objects, scenes, and states. The split was further shuffled and divided into *Train*/*Val*/*Test* sets proportioned at 70%/15%/15%. Notably, in addition to providing valid initialization configurations, demonstrations for evaluation splits may be used to provide intermediate ground truth for diagnosing model performance (Sec. 4.3). Furthermore, we created the *Generalization* splits *Novel Object/Scene/State* by gathering data with one of the three components (*i.e.*, objects, scenes, and goal states) unseen; *e.g.*, the *Novel Object* split comprises data of unseen objects, and seen scenes and states.

While the *Novel State* split addresses the generalization of unseen goal states, we expect that grounding on continuous state representations should help the agent to adapt to any arbitrary state within a continuous range. Therefore, we make the *Any State* split with seen objects and scenes, setting the goal states uniformly distributed over a continuous range, *e.g.*, 0%–100%. Such a design resembles universal tasks with arbitrary goal states and facilitates the evaluation of state generalization. Tab. 3 presents the data statistics.

Metrics. A task instance is regarded as a success when the success condition is satisfied continually for 2 seconds. The success condition requires the current state to be within a tolerance threshold from the goal state; *i.e.*, the success range. The tolerances are derived according to human behaviors and are shown in Tab. 2. Note that TRANSFERWATER imposes the extra condition that only 10% or less of the water can be spilled. The execution of evaluation resembles the composition of sub-task stages in the motion planner (details in Appendix B of [27]). To avoid accidental triggering, we check the success condition after the agent completes the final stage. For example, in the task "*pour half of the water out of the cup*", the agent succeeds if $40\% \sim 60\%$ of the water remains in the cup for 2 seconds after the agent has reoriented the cup upright. We have adopted success rate as the evaluation metric in the ARNOLD.

4. Experiments

4.1. Experimental Setup

Models. To evaluate the existing language-conditioned robotic manipulation models on ARNOLD, we chose two state-of-the-art models as our primary focus: 6D-CLIPort [89] and PerAct [71].

- 6D-CLIPort takes as input an RGB-D image from the top-down view and predicts end-effector poses for the current object and the target action. Each end-effector pose contains an action translation and a categorical prediction over discretized Euler angles. 6D-CLIPort comprises three branches to process the multi-modal input: Transporter-ResNet [87] for the spatial stream, CLIP visual encoder and language encoder [66] for the semantic stream.
- PerAct takes RGB-D images as input to fuse a 3D voxelized grid. In addition, PerAct also requires the proprioception, including gripper states and the current timestep. The proprioception features are tiled on the voxel grid. Next, the hybrid voxel grid is downsampled and flattened to a sequence. Meanwhile, the language instruction is fed to a language encoder (*e.g.*, CLIP [66]) and then appended to the sequence. PerAct uses Perceiver-IO [34] to resample a compact latent representation from the multi-modal long sequence. After decoding, PerAct finally outputs a Q function over the original voxel grid for the prediction of action translation. Similar to 6D-CLIPort, PerAct also outputs a categorical distribution over discretized Euler angles for the prediction of action rotation. In contrast to the implementation in [71], we discard the heads for predicting gripper and collision. Instead, we add an optional head for state prediction.

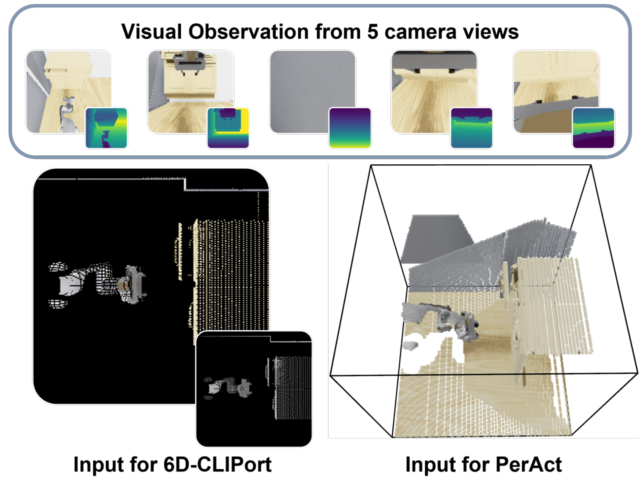

Figure 3. Visualization of the input representations of models. The five cameras are from the front, base, left, and two wrist views. The left camera view is occluded by the wall. Finally, these camera views fuse into an RGB-D image for 6D-CLIPort and a voxel grid for PerAct.

Moreover, we considered three model variants of PerAct in our experiments: (1) PerAct without language (PerAct w/o L) for studying the importance of language-grounding, (2) PerAct with additional supervision on state value (PerAct†) to show the urgency of state modeling for tasks in ARNOLD, and (3) PerAct trained in the multi-task setting (PerAct MT) given the great potential of multi-task learning shown in [71]. For PerAct†, we provide additional state supervision by adding an extra output head to regress the normalized state values from the hidden features. We also tried other manipulation models; *e.g.*, BC-Z in Appendix D of [27].

Implementation Details. We obtain the visual representations for the models based on the five rendered RGB-D images as follows: With the camera parameters, we cast the pixels back to 3D and thus derive a point cloud for each view. With these point clouds in 3D scenes, we apply a perception bounding box to make the keypoint learning more tractable. In our setting, the cube spans 126 cm on each axis and the cube center (p_x, p_y, p_z) is 50 cm away from the robot base along the robot's forward direction. Next, we obtain the visual representations (visualized in Fig. 3) as follows:

- For 6D-CLIPort, with each pixel occupying a size of 0.56 cm, we can map the 126 cm $\times$ 126 cm perceived area to a 224×224 top-down view image. We project the point clouds with their color and distance information onto this

Table 4. Evaluation results of the models on various tasks and splits, measured by success rate and shown in percentages. The gray figures indicate performances with the first-phase ground truth. For each model, the first row shows the performance on the *Test* set, and the following three rows show those on the *Novel* splits of *Object*, *Scene*, and *State*. The last row indicates the performances on the *Any State* split. Tasks are abbreviated for more space. Average performances on eight tasks are appended to each row. **w/o L**: without language instruction. †: model variants with state modeling. **MT**: multi-task models.

	P.Object		R.Object		O.Drawer		C.Drawer		O.Cabinet		C.Cabinet		P.Water		T.Water		Average	
6D-CLIPort	6.72	25.37	0.00	0.00	0.00	0.00	0.00	2.70	0.00	0.00	0.00	5.83	0.00	0.00	0.00	7.14	0.84	5.13
Object	8.36	28.36	0.00	0.00	0.00	0.00	0.00	0.40	0.00	1.23	0.00	1.82	0.00	0.00	0.00	3.92	1.05	4.47
Scene	10.41	24.43	0.00	0.00	0.00	1.57	0.00	0.00	0.00	0.55	1.27	1.27	0.00	0.00	0.00	12.00	1.46	4.98
State	0.00	0.00	0.00	0.00	0.00	0.57	0.75	1.13	0.00	0.83	0.00	2.78	0.00	0.00	0.00	16.81	0.09	2.77
Any State	10.45	29.10	1.30	2.60	0.84	0.00	0.68	1.35	0.00	5.80	0.00	2.91	0.00	1.49	0.00	7.14	1.66	6.30
PerAct (w/o L)	25.37	33.58	14.29	7.79	17.65	36.13	47.30	52.03	8.70	34.78	7.77	10.68	14.93	11.94	5.36	14.29	17.67	25.15
Object	29.09	26.55	8.77	3.51	3.87	20.00	24.70	**32.67**	0.00	0.00	**1.82**	**7.27**	16.67	29.17	9.80	**19.61**	11.84	17.35
Scene	26.70	24.89	14.63	14.63	19.61	33.33	48.15	54.32	**1.10**	3.87	1.91	1.27	13.79	20.69	6.00	16.00	16.49	21.13
State	0.34	0.00	0.00	0.95	9.20	8.05	1.70	2.08	0.00	2.07	1.39	1.39	1.08	2.69	5.04	**8.40**	2.34	3.20
Any State	20.15	19.40	12.99	12.99	13.45	30.25	20.95	24.32	**5.80**	**26.09**	**14.56**	**15.53**	14.93	16.42	1.79	7.14	13.08	19.02
PerAct	94.03	**97.76**	19.48	24.68	31.09	44.54	**60.81**	66.22	**24.64**	42.03	22.33	45.63	**55.22**	74.63	**32.14**	46.43	42.47	55.24
Object	86.55	**92.73**	**11.40**	**35.09**	**6.45**	21.29	**26.29**	27.89	0.00	0.00	**1.82**	5.45	**36.46**	**42.71**	**13.73**	13.73	**22.84**	**29.86**
Scene	**72.85**	**84.62**	**17.07**	31.71	20.78	31.37	**66.67**	64.20	0.00	**4.97**	5.10	19.11	**33.33**	51.72	22.00	**36.00**	**29.73**	40.46
State	2.38	0.68	0.00	0.95	**10.92**	12.64	**13.77**	17.74	0.00	5.81	1.39	1.39	1.61	1.08	**5.88**	1.68	4.49	5.25
Any State	47.01	**50.75**	7.79	**19.48**	**21.85**	30.25	18.92	25.00	**5.80**	21.74	3.88	**15.53**	14.93	25.37	10.71	14.29	16.36	25.30
PerAct†	**94.78**	95.52	**24.68**	**28.57**	**36.13**	**52.94**	60.14	**68.24**	23.19	**49.28**	**30.10**	**48.54**	49.25	**85.07**	28.57	**53.57**	**43.36**	**60.22**
Object	**87.27**	91.27	10.53	32.46	1.94	**22.58**	18.73	25.10	0.00	**4.94**	0.00	5.45	34.38	33.33	9.80	11.76	20.33	28.36
Scene	69.68	84.16	13.41	37.80	**25.49**	**38.43**	60.49	**67.90**	0.55	**4.97**	6.37	**19.75**	29.89	**63.22**	**26.00**	24.00	28.99	**42.53**
State	0.68	2.38	0.48	0.00	10.06	12.93	13.58	**18.11**	0.00	**6.22**	0.00	8.33	2.15	1.61	**5.88**	2.52	4.10	6.51
Any State	**48.51**	47.76	**14.29**	14.29	21.01	**33.61**	**23.65**	**28.38**	4.35	24.64	6.80	13.59	**26.87**	**31.34**	**14.29**	**19.64**	**19.97**	**26.66**
PerAct (MT)	88.81	88.81	3.90	22.08	26.05	43.70	33.78	52.03	11.59	33.33	20.39	37.86	34.33	58.21	14.29	23.21	29.14	44.90
Object	77.09	77.45	7.02	20.18	1.29	15.48	13.55	25.50	0.00	0.00	**1.82**	**7.27**	13.54	30.21	1.96	11.76	14.53	23.48
Scene	68.78	77.83	10.98	**41.46**	13.33	25.88	29.63	55.56	0.00	**4.97**	5.73	9.55	19.54	43.68	4.00	16.00	19.00	34.37
State	**9.18**	**12.59**	0.00	1.90	8.05	12.07	9.43	13.40	0.00	2.07	1.39	0.00	2.69	6.45	**5.88**	5.88	**4.58**	6.80
Any State	36.57	45.52	5.19	10.39	15.13	19.33	16.22	21.62	1.45	8.70	0.97	3.88	8.96	14.93	3.57	17.86	11.01	17.78
PerAct (MT)†	90.30	92.54	14.29	20.78	25.21	47.90	33.78	56.76	20.29	39.13	19.42	37.86	26.87	64.18	17.86	30.36	31.00	48.69
Object	81.09	85.45	7.89	24.56	3.23	**22.58**	14.74	28.69	0.00	**4.94**	**1.82**	5.45	9.38	20.83	3.92	**19.61**	15.26	26.51
Scene	67.87	79.64	7.32	29.27	12.94	25.49	39.51	65.43	**1.10**	4.42	**7.01**	14.01	10.34	43.68	8.00	22.00	19.26	35.49
State	2.04	3.06	**0.95**	**2.38**	9.20	**18.68**	6.98	11.13	0.00	3.73	**2.78**	**11.11**	**6.45**	**9.14**	1.68	4.20	3.76	**7.93**
Any State	46.27	47.01	12.99	12.99	12.61	23.53	14.86	26.35	4.35	5.80	4.85	8.74	16.42	25.37	3.57	5.36	14.49	19.39

image. Note that the distance represents height rather than depth. If a pixel is related to multiple points (occlusion), the RGB values will be the average of these points and the distance will be the largest height.

- For PerAct, we set the size of voxel grid to be 120^3, with each voxel covering the 3D context in a $1.05\,\mathrm{cm}^3$ volume. We then aggregate point clouds of all views into the voxel grid, including coordinates, RGB, positional embedding, and occupancy.

Please check https://github.com/arnold-benchmark/arnold for detailed implementations of the models and their variants.

Learning. We trained all the models on the *Train* split and performed model selection among ten checkpoints on the *Val* split. Following common practice [89, 71], we used waypoints in demonstrations to facilitate model learning. Specifically, we unify the execution of tasks in ARNOLD into a two-phase procedure: grasping the target object and then manipulating it toward the goal state. We followed the training settings of [89, 71]. Additionally, in multi-task training we sampled each training data by first uniformly sampling the task, then sampling a demonstration of the corresponding task. We set the number of training iterations to 100k for the single-task and 200k for the multi-task setting. We performed all the training on a single NVIDIA A100 GPU with batch size 8.

Evaluation Execution. The evaluation executor of each task resembles the pipeline of a motion planner yet has a slight difference (see details in Appendix B of [27]). The model's predictions are converted back to keypoints of subtask stages for evaluation execution. We make our evaluation strict and appropriate by avoiding shortcuts on the object state during robot motion. For example, the task instructing "*pull the cabinet* 50% *open*" will not be considered a success even if the cabinet is held around 50% open for 2 seconds so long as the motion planner has not executed its final action. To eliminate the influence of a first-phase failure (*i.e.*, failing to grasp the target object or object part), we conduct additional evaluations that provide first-phase ground truth.

4.2. Experimental Results

We report experimental results in Tab. 4 (black) and present our findings and analyses below.

Across Models. Comparing the baseline models 6D-CLIPort, PerAct, and PerAct (MT), we found that 6D-

CLIPort fails on most of the tasks. We conjecture that this stems from (1) the information lost in the input representation when compressing complex 3D scenes into a single image and (2) the difficulty in regressing target height values for action translation. By contrast, the voxelized representations in PerAct provide rich 3D contexts that benefit model learning. With such differences, PerAct outperforms 6D-CLIPort significantly. Meanwhile, we observed performance drops for PerAct (MT) compared to PerAct on all tasks. This indicates that it is still difficult to leverage more diverse multi-task data for efficiently learning better task policies in ARNOLD, especially given its challenges in both grounding and manipulation.

Across Tasks. The most challenging tasks in ARNOLD are REORIENTOBJECT and {OPEN,CLOSE}CABINET. REORIENTOBJECT is difficult because it involves estimation of the bottle orientations, and models can often be confused by visually similar states. For example, the action for the goal state of 45° will lead to a goal state of 135° if the bottle orientation is reversed. Manipulating a cabinet proved to be challenging even with privileged information to specify goals [59, 29] as it requires accurate prediction of both interacting position, rotation, and precise continuous motion control. Replacing privileged information with instructions will only make it harder. Furthermore, we observed superior model performance in POURWATER compared to TRANSFERWATER. This is because transferring water requires position alignment between cups to avoid spillage.

On Generalization Splits. In general, we observed performance drops for most models when transfering to *Novel* generalization splits, especially on the *Novel State* split. This reveals that, without proper modeling of continuous states, generalizing the grounding of seen goal states to unseen ones remains challenging. Meanwhile, the performance drop on the *Novel Object* and *Novel Scene* splits varies according to the tasks. For tasks where the objects occupy substantial space (*e.g.*, drawer), the impact of unseen objects is more significant than unseen scenes.

For the *Any State* split, the models' performances were inferior compared with the *Novel Object/Scene* split and superior to those on the *Novel State* split. As goal states are uniformly sampled from a continuous spectrum, the success ranges of seen goal states are likely to cover a large portion of the spectrum, making the *Any State* generalization interpolations of learned knowledge and skills. This suggests an interesting research question that can be investigated with ARNOLD: How can the task learning model better generalize by interpolating within ranges and extrapolating to out-of-range goal states?

Remarks. The key findings from our experiments with ARNOLD are as follows:

- Current models still struggle with tasks that require complex manipulation skills (*e.g.*, manipulating cabinets). *This heightens the demand for better policy learning models to tackle challenging manipulation tasks.*
- The low success rate of models on all generalization splits motivates the necessity for (1) *increasingly fine-grained representations for perceptual inputs*, (2) *finer modeling of continuous object states*, and (3) *better alignment between language and robot actions.*
- The *Any state* experiments suggest that *state generalization could potentially be achieved through the interpolation of acquired knowledge and skills*. This promotes approaches with deeper insights into systematic generalization for robot skill adaptation.

4.3. Ablation Studies

Influence of Language. PerAct (w/o L), trained with a single-task scheme, exhibits a considerable performance gap behind PerAct. This indicates the importance of the goal-state information in ARNOLD. Meanwhile, we observe a relatively small gap for REORIENTOBJECT. We believe this is due to (1) the bottleneck of this task lying in the ambiguity of visual perception, as discussed in Sec. 4.2 and (2) the difficulty of grounding angles from current visual observations. On the other hand, the significant performance gap on PICKUPOBJECT shows the effectiveness of language grounding in visually more identifiable concepts such as translation distance.

Importance of State Modeling. The PerAct variants with state supervision (†) were expected to realize a performance gain from explicit state prediction supervision. However, we observed marginal improvements on the *Test* split and limited enhancements on generalization splits for this method. This indicates that such end-to-end state supervision is insufficient for state modeling in ARNOLD and calls for better approaches to representing and modeling the continuous object states in robotic task learning.

With Intermediate Oracle. To better demonstrate how well models understand goal states, we provided models with first-phase ground truth (*i.e.*, grasping positions) and report their performance in Tab. 4 (gray). As expected, we observed significant performance gains with such ground truth. Moreover, directly comparing the scores with the first-phase oracle, we can also observe larger gaps between PerAct and PerAct†, as well as PerAct (MT) and PerAct (MT)†. These results demonstrate that explicit state modeling is indeed beneficial for goal state understanding.

Choice of Language Encoder. We investigated model ablation over the language encoder by switching the default language encoder CLIP [66] in PerAct to T5-base [67]. Due to space constraints, we report model performance only on OPENDRAWER. As shown in Fig. 4, PerAct-T5 outperforms PerAct-CLIP on all the benchmarking splits. This may be due to the inefficacy of the global language representation learned in CLIP in representing fine-grained goal states for

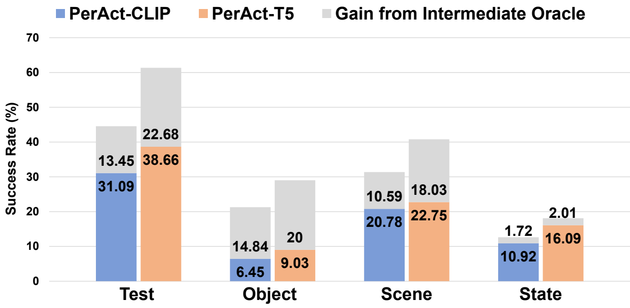

Figure 4. Model ablation results with different language encoders.

precise control. By contrast, T5 is a general-purpose language processing model that offers the ability to maintain detailed information through language modeling and generation. This shows that finer language embeddings may benefit the language grounding of robots to some extent.

4.4. Sim2Real Transfer Experiment

As a realistic simulation environment, one key question to address is: *To what extent can the agents trained in* `ARNOLD` *generalize to real-world scenarios*? To this end, we set up a real-world environment for testing the Sim2Real transfer capabilities of agents. Specifically, we used the Franka robot arm to manipulate previously unseen real-world objects with partial point cloud observations captured through a single RGB-D camera from the left view (Fig. 2). We experimented with PerAct, which was trained in `ARNOLD` to open/close 2 different drawers and pick up 5 different objects. We mitigated the Sim2Real gap as follows:

1. Perception: We used high-fidelity rendering. Due to the imperfect depth sensor in the real world, we sprayed contrast aiding paint onto metallic and transparent areas. For diffuse objects with acceptable depth quality, further domain adaptation would be beneficial.
2. Control: Instead of using the qpos control API, we used an inverse kinematics (IK) controller to perform real-robot actions, which avoids error accumulation.

Additional experimental details can be found in Appendix D of [27]. Throughout our experiments, we observed that models trained in `ARNOLD` show preliminary Sim2Real transfer capabilities; *i.e.*, reasonable predictions for both picking up objects and manipulating drawers, as shown in Fig. 5. However, the actual robot manipulation continues to struggle because of the Sim2Real gap. For example, when opening a non-plastic drawer in the real world, the robot encounters high friction and is therefore susceptible to prediction errors that lead to inexecutable actions (*e.g.*, exceeding the critical friction angle). With more fine-grained object assets, we believe that the flexible design of the `ARNOLD` simulator can gradually close this Sim2Real gap by providing more realistic simulations.

Figure 5. Real-world experiments with inference results shown on the upper right. The red dots indicate the predicted positions of the next action for the end-effector.

5. Conclusion

We have presented `ARNOLD`, a benchmark for language-grounded task learning in realistic 3D interactive environments with diverse scenes, objects, and continuous object states. We devised a systematic benchmark comprising eight challenging language-grounded robot tasks and evaluation splits for robot skill generalization in novel scene, object, and goal-state scenarios. We conducted extensive experiments and analyses to pinpoint the limitations of the current models and identified promising research directions for grounded task learning.

Limitations and Future Work. (1) Despite our focus on realistic simulation, the gap between `ARNOLD` and real-world scenarios still remains. However, with our flexible design, we can reduce this gap by adding more realistic variations of both scenes and objects [25] to the assets library. (2) Current tasks in `ARNOLD` do not require much high-level planning knowledge. However, as we found in experiments, short-horizon fine-grained control has not been well solved yet. This defers long-horizon tasks to the future. (3) Despite the diversity we inject in language, the template-based generation inevitably restricts language variations. As a key component for generalization, it is critical to extend `ARNOLD` with richer language instructions (*e.g.*, by prompting LLMs) in future work. (4) Current imitation learning models rely on prior simplifications (*e.g.*, the two-phase learning) to learn sparse keypoints. Limited by the amount and diversity of expert policies, we need improved methods for modeling continuous object states and learning generalizable policies from scarce data. (5) The realistic and resourceful environment provided by `ARNOLD` can also facilitate the pursuit of versatile agents with comprehensive capabilities [38, 31]. (6) Scaling up demonstrations is another crucial future step, which can induce stronger capabilities [4, 3].

Acknowledgments

This work is supported in part by the National Key R&D Program of China (2021ZD0150200) and NVIDIA GPU Grant. In addition, we thank Chiao Lu for the help with the environment setup for large-scale experiments.

References

[1] P. Anderson, Q. Wu, D. Teney, J. Bruce, M. Johnson, N. Sünderhauf, I. Reid, S. Gould, and A. Van Den Hengel, "Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2

[2] D. Batra, A. X. Chang, S. Chernova, A. J. Davison, J. Deng, V. Koltun, S. Levine, J. Malik, I. Mordatch, R. Mottaghi *et al.*, "Rearrangement: A challenge for embodied AI," *arXiv preprint arXiv:2011.01975*, 2020. 2

[3] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, J. Dabis, C. Finn, K. Gopalakrishnan, K. Hausman, A. Herzog, J. Hsu *et al.*, "RT-1: Robotics transformer for real-world control at scale," *arXiv preprint arXiv:2212.06817*, 2022. 9

[4] A. Brohan, Y. Chebotar, C. Finn, K. Hausman, A. Herzog, D. Ho, J. Ibarz, A. Irpan, E. Jang, R. Julian *et al.*, "Do as I can, not as I say: Grounding language in robotic affordances," in *Conference on Robot Learning (CoRL)*, 2022. 3, 9

[5] A. Chang, A. Dai, T. Funkhouser, M. Halber, M. Niessner, M. Savva, S. Song, A. Zeng, and Y. Zhang, "Matterport3D: Learning from RGB-D data in indoor environments," in *International Conference on 3D Vision (3DV)*, 2017. 2

[6] C. Chen, C. Schissler, S. Garg, P. Kobernik, A. Clegg, P. Calamia, D. Batra, P. W. Robinson, and K. Grauman, "SoundSpaces 2.0: A simulation platform for visual-acoustic learning," in *Advances in Neural Information Processing Systems Datasets and Benchmarks Track (NeurIPS Datasets and Benchmarks)*, 2022. 2

[7] C.-A. Cheng, M. Mukadam, J. Issac, S. Birchfield, D. Fox, B. Boots, and N. Ratliff, "RMPflow: A computational graph for automatic motion policy generation," in *Algorithmic Foundations of Robotics XIII: Proceedings of the 13th Workshop on the Algorithmic Foundations of Robotics 13*. Springer, 2020, pp. 441–457. 5

[8] A. Das, S. Datta, G. Gkioxari, S. Lee, D. Parikh, and D. Batra, "Embodied question answering," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2

[9] M. Deitke, D. Batra, Y. Bisk, T. Campari, A. X. Chang, D. S. Chaplot, C. Chen, C. P. D'Arpino, K. Ehsani, A. Farhadi *et al.*, "Retrospectives on the embodied AI workshop," *arXiv preprint arXiv:2210.06849*, 2022. 2

[10] M. Deitke, W. Han, A. Herrasti, A. Kembhavi, E. Kolve, R. Mottaghi, J. Salvador, D. Schwenk, E. VanderBilt, M. Wallingford *et al.*, "RoboTHOR: An open simulation-to-real embodied AI platform," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2

[11] M. Deitke, E. VanderBilt, A. Herrasti, L. Weihs, J. Salvador, K. Ehsani, W. Han, E. Kolve, A. Farhadi, A. Kembhavi *et al.*, "ProcTHOR: Large-scale embodied AI using procedural generation," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. 2

[12] Y. Deng, D. Guo, X. Guo, N. Zhang, H. Liu, and F. Sun, "MQA: Answering the question via robotic manipulation," in *Robotics: Science and Systems (RSS)*, 2020. 3

[13] Y. Di, R. Zhang, Z. Lou, F. Manhardt, X. Ji, N. Navab, and F. Tombari, "GPV-Pose : Category-level object pose estimation via geometry-guided point-wise voting," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 3

[14] D. Driess, F. Xia, M. S. M. Sajjadi, C. Lynch, A. Chowdhery, B. Ichter, A. Wahid, J. Tompson, Q. Vuong, T. Yu, W. Huang, Y. Chebotar, P. Sermanet, D. Duckworth, S. Levine, V. Vanhoucke, K. Hausman, M. Toussaint, K. Greff, A. Zeng, I. Mordatch, and P. Florence, "PaLM-E: An embodied multimodal language model," in *arXiv preprint arXiv:2303.03378*, 2023. 3

[15] J. Duan, S. Yu, H. L. Tan, H. Zhu, and C. Tan, "A survey of embodied AI: From simulators to research tasks," *IEEE Transactions on Emerging Topics in Computational Intelligence*, 2022. 2

[16] K. Ehsani, W. Han, A. Herrasti, E. VanderBilt, L. Weihs, E. Kolve, A. Kembhavi, and R. Mottaghi, "ManipulaTHOR: A framework for visual object manipulation," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 2

[17] H. Fu, W. Xu, H. Xue, H. Yang, R. Ye, Y. Huang, Z. Xue, Y. Wang, and C. Lu, "RFUniverse: A physics-based action-centric interactive environment for everyday household tasks," *arXiv preprint arXiv:2202.00199*, 2022. 2

[18] H. Fu, B. Cai, L. Gao, L.-X. Zhang, J. Wang, C. Li, Q. Zeng, C. Sun, R. Jia, B. Zhao *et al.*, "3D-FRONT: 3D furnished rooms with layouts and semantics," in *International Conference on Computer Vision (ICCV)*, 2021. 2, 4

[19] C. Gan, J. Schwartz, S. Alter, D. Mrowca, M. Schrimpf, J. Traer, J. De Freitas, J. Kubilius, A. Bhandwaldar, N. Haber *et al.*, "ThreeDWorld: A platform for interactive multi-modal physical simulation," in *Advances in Neural Information Processing Systems Datasets and Benchmarks Track (NeurIPS Datasets and Benchmarks)*, 2021. 2, 3

[20] C. Gan, S. Zhou, J. Schwartz, S. Alter, A. Bhandwaldar, D. Gutfreund, D. L. Yamins, J. J. DiCarlo, J. McDermott, A. Torralba *et al.*, "The ThreeDWorld transport challenge: A visually guided task-and-motion planning benchmark towards physically realistic embodied AI," in *International Conference on Robotics and Automation (ICRA)*. IEEE, 2022. 3

[21] Q. Gao, M. Doering, S. Yang, and J. Chai, "Physical causality of action verbs in grounded language understanding," in *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2016. 2

[22] X. Gao, Q. Gao, R. Gong, K. Lin, G. Thattai, and G. S. Sukhatme, "DialFRED: Dialogue-enabled agents for embodied instruction following," *IEEE Robotics and Automation Letters (RA-L)*, 2022. 2

[23] X. Gao, R. Gong, T. Shu, X. Xie, S. Wang, and S.-C. Zhu, "VRKitchen: an interactive 3D virtual environment for task-oriented learning," *arXiv preprint arXiv:1903.05757*, 2019. 3

[24] H. Geng, Z. Li, Y. Geng, J. Chen, H. Dong, and H. Wang, "PartManip: Learning cross-category generalizable part manipulation policy from point cloud observations," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 2

[25] H. Geng, H. Xu, C. Zhao, C. Xu, L. Yi, S. Huang, and H. Wang, "GAPartNet: Cross-category domain-generalizable object perception and manipulation via generalizable and actionable parts," *arXiv preprint arXiv:2211.05272*, 2022. 9

[26] Y. Geng, B. An, H. Geng, Y. Chen, Y. Yang, and H. Dong, "End-to-end affordance learning for robotic manipulation," *arXiv preprint arXiv:2209.12941*, 2022. 2

[27] R. Gong, J. Huang, Y. Zhao, H. Geng, X. Gao, Q. Wu, W. Ai, Z. Zhou, D. Terzopoulos, S.-C. Zhu, B. Jia, and S. Huang, "ARNOLD: A benchmark for language-grounded task learning with continuous states in realistic 3D scenes," *arXiv preprint arXiv:2304.04321*, 2023. 4, 5, 6, 7, 9

[28] J. Gu, D. S. Chaplot, H. Su, and J. Malik, "Multi-skill mobile manipulation for object rearrangement," *International Conference on Learning Representations (ICLR)*, 2022. 2

[29] J. Gu, F. Xiang, X. Li, Z. Ling, X. Liu, T. Mu, Y. Tang, S. Tao, X. Wei, Y. Yao *et al.*, "ManiSkill2: A unified benchmark for generalizable manipulation skills," in *International Conference on Learning Representations (ICLR)*, 2023. 3, 5, 8

[30] Y. Hong, Z. Wang, Q. Wu, and S. Gould, "Bridging the gap between learning in discrete and continuous environments for vision-and-language navigation," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 2

[31] J. Huang, W. Y. Zhu, B. Jia, Z. Wang, X. Ma, Q. Li, and S. Huang, "Perceive, ground, reason, and act: A benchmark for general-purpose visual representation," *arXiv preprint arXiv:2211.15402*, 2022. 9

[32] S. Huang, Z. Wang, P. Li, B. Jia, T. Liu, Y. Zhu, W. Liang, and S.-C. Zhu, "Diffusion-based generation, optimization, and planning in 3D scenes," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 3

[33] W. Huang, F. Xia, T. Xiao, H. Chan, J. Liang, P. Florence, A. Zeng, J. Tompson, I. Mordatch, Y. Chebotar *et al.*, "Inner monologue: Embodied reasoning through planning with language models," in *Conference on Robot Learning (CoRL)*, 2022. 3

[34] A. Jaegle, S. Borgeaud, J.-B. Alayrac, C. Doersch, C. Ionescu, D. Ding, S. Koppula, D. Zoran, A. Brock, E. Shelhamer *et al.*, "Perceiver IO: A general architecture for structured inputs & outputs," in *International Conference on Learning Representations (ICLR)*, 2022. 6

[35] U. Jain, L. Weihs, E. Kolve, A. Farhadi, S. Lazebnik, A. Kembhavi, and A. Schwing, "A cordial sync: Going beyond marginal policies for multi-agent embodied tasks," in *European Conference on Computer Vision (ECCV)*, 2020. 2

[36] U. Jain, L. Weihs, E. Kolve, M. Rastegari, S. Lazebnik, A. Farhadi, A. G. Schwing, and A. Kembhavi, "Two body problem: Collaborative visual task completion," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2

[37] S. James, Z. Ma, D. R. Arrojo, and A. J. Davison, "RLBench: The robot learning benchmark & learning environment," *IEEE Robotics and Automation Letters (RA-L)*, 2020. 2, 3

[38] Y. Jiang, A. Gupta, Z. Zhang, G. Wang, Y. Dou, Y. Chen, L. Fei-Fei, A. Anandkumar, Y. Zhu, and L. Fan, "VIMA: General robot manipulation with multimodal prompts," *arXiv preprint arXiv:2210.03094*, 2022. 3, 9

[39] A. Kamath, M. Singh, Y. LeCun, G. Synnaeve, I. Misra, and N. Carion, "MDETR-modulated detection for end-to-end multi-modal understanding," in *International Conference on Computer Vision (ICCV)*, 2021. 2

[40] E. Kolve, R. Mottaghi, W. Han, E. VanderBilt, L. Weihs, A. Herrasti, D. Gordon, Y. Zhu, A. Gupta, and A. Farhadi, "AI2-THOR: An interactive 3D environment for visual ai," *arXiv preprint arXiv:1712.05474*, 2017. 2, 4

[41] J. Krantz, E. Wijmans, A. Majumdar, D. Batra, and S. Lee, "Beyond the nav-graph: Vision-and-language navigation in continuous environments," in *European Conference on Computer Vision (ECCV)*, 2020. 2

[42] V. Kumar and E. Todorov, "MuJoCo HAPTIX: A virtual reality system for hand manipulation," in *2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*, 2015. 2

[43] C. Li, F. Xia, R. Martín-Martín, M. Lingelbach, S. Srivastava, B. Shen, K. Vainio, C. Gokmen, G. Dharan, T. Jain *et al.*, "iGibson 2.0: Object-centric simulation for robot learning of everyday household tasks," in *Conference on Robot Learning (CoRL)*, 2021. 2

[44] C. Li, R. Zhang, J. Wong, C. Gokmen, S. Srivastava, R. Martín-Martín, C. Wang, G. Levine, M. Lingelbach, J. Sun *et al.*, "BEHAVIOR-1K: A benchmark for embodied AI with 1,000 everyday activities and realistic simulation," in *Conference on Robot Learning (CoRL)*, 2022. 2, 3

[45] X. Lin, Y. Wang, J. Olkin, and D. Held, "SoftGym: Benchmarking deep reinforcement learning for deformable object manipulation," in *Conference on Robot Learning (CoRL)*, 2020. 2, 3

[46] M. Liu, X. Li, Z. Ling, Y. Li, and H. Su, "Frame mining: a free lunch for learning robotic manipulation from 3D point clouds," in *Conference on Robot Learning (CoRL)*, 2022. 4

[47] Y. Liu, P. Wei, and S.-C. Zhu, "Jointly recognizing object fluents and tasks in egocentric videos," in *International Conference on Computer Vision (ICCV)*, 2017. 3

[48] W. E. Lorensen and H. E. Cline, "Marching cubes: A high resolution 3D surface construction algorithm," *ACM SIGGRAPH Computer Graphics*, vol. 21, no. 4, pp. 163–169, 1987. 4

[49] J. Lu, D. Batra, D. Parikh, and S. Lee, "ViLBERT: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks," *Advances in Neural Information Processing Systems (NeurIPS)*, 2019. 2

[50] C. Lynch and P. Sermanet, "Language conditioned imitation learning over unstructured data," *Robotics: Science and Systems (RSS)*, 2021. 3

[51] C. Lynch, A. Wahid, J. Tompson, T. Ding, J. Betker, R. Baruch, T. Armstrong, and P. Florence, "Interactive language: Talking to robots in real time," *arXiv preprint arXiv:2210.06407*, 2022. 3

[52] L. Ma, J. Meng, S. Liu, W. Chen, J. Xu, and R. Chen, "Sim2Real^2: Actively building explicit physics model for precise articulated object manipulation," in *International Conference on Robotics and Automation (ICRA)*, 2023. 3

[53] X. Ma, S. Yong, Z. Zheng, Q. Li, Y. Liang, S.-C. Zhu, and S. Huang, "SQA3D: Situated question answering in 3D scenes," in *International Conference on Learning Representations (ICLR)*, 2022. 2

[54] M. Macklin and M. Müller, "Position based fluids," *ACM Transactions on Graphics (TOG)*, vol. 32, no. 4, pp. 1–12, 2013. 4

[55] V. Makoviychuk, L. Wawrzyniak, Y. Guo, M. Lu, K. Storey, M. Macklin, D. Hoeller, N. Rudin, A. Allshire, A. Handa *et al.*, "Isaac Gym: High performance GPU-based physics simulation for robot learning," *Advances in Neural Information Processing Systems Datasets and Benchmarks Track (NeurIPS Datasets and Benchmarks)*, 2021. 4

[56] O. Mees, L. Hermann, and W. Burgard, "What matters in language conditioned robotic imitation learning over unstructured data," *IEEE Robotics and Automation Letters (RA-L)*, 2022. 2

[57] O. Mees, L. Hermann, E. Rosete-Beas, and W. Burgard, "CALVIN: A benchmark for language-conditioned policy learning for long-horizon robot manipulation tasks," *IEEE Robotics and Automation Letters (RA-L)*, 2022. 2, 3

[58] M. Mittal, C. Yu, Q. Yu, J. Liu, N. Rudin, D. Hoeller, J. L. Yuan, P. P. Tehrani, R. Singh, Y. Guo *et al.*, "ORBIT: A unified simulation framework for interactive robot learning environments," *arXiv preprint arXiv:2301.04195*, 2023. 3

[59] T. Mu, Z. Ling, F. Xiang, D. Yang, X. Li, S. Tao, Z. Huang, Z. Jia, and H. Su, "ManiSkill: Generalizable manipulation skill benchmark with large-scale demonstrations," *Advances in Neural Information Processing Systems Datasets and Benchmarks Track (NeurIPS Datasets and Benchmarks)*, 2021. 3, 4, 5, 8

[60] T. Nagarajan and K. Grauman, "Attributes as operators: factorizing unseen attribute-object compositions," in *European Conference on Computer Vision (ECCV)*, 2018. 3

[61] ——, "Learning affordance landscapes for interaction exploration in 3D environments," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. 2

[62] S. Nair, E. Mitchell, K. Chen, S. Savarese, C. Finn *et al.*, "Learning language-conditioned robot behavior from offline data and crowd-sourced annotation," in *Conference on Robot Learning (CoRL)*, 2022. 3

[63] A. Padmakumar, J. Thomason, A. Shrivastava, P. Lange, A. Narayan-Chen, S. Gella, R. Piramuthu, G. Tur, and D. Hakkani-Tur, "TEACh: Task-driven embodied agents that chat," in *AAAI Conference on Artificial Intelligence (AAAI)*, 2022. 2

[64] A. Pashevich, C. Schmid, and C. Sun, "Episodic transformer for vision-and-language navigation," in *International Conference on Computer Vision (ICCV)*, 2021. 2

[65] X. Puig, K. Ra, M. Boben, J. Li, T. Wang, S. Fidler, and A. Torralba, "VirtualHome: Simulating household activities via programs," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2

[66] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark *et al.*, "Learning transferable visual models from natural language supervision," in *International Conference on Machine Learning (ICML)*, 2021. 2, 6, 8

[67] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," *Journal of Machine Learning Research (JMLR)*, 2020. 8

[68] C. Saharia, W. Chan, S. Saxena, L. Li, J. Whang, E. Denton, S. K. S. Ghasemipour, B. K. Ayan, S. S. Mahdavi, R. G. Lopes *et al.*, "Photorealistic text-to-image diffusion models with deep language understanding," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. 2

[69] P. Shirley and R. K. Morley, *Realistic Ray Tracing*. AK Peters, Ltd., 2008. 4

[70] M. Shridhar, L. Manuelli, and D. Fox, "CLIPort: What and where pathways for robotic manipulation," in *Conference on Robot Learning (CoRL)*, 2022. 2, 3

[71] ——, "Perceiver-Actor: A multi-task transformer for robotic manipulation," *Conference on Robot Learning (CoRL)*, 2022. 2, 3, 4, 6, 7

[72] M. Shridhar, J. Thomason, D. Gordon, Y. Bisk, W. Han, R. Mottaghi, L. Zettlemoyer, and D. Fox, "ALFRED: A benchmark for interpreting grounded instructions for everyday tasks," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2, 3

[73] S. Srivastava, C. Li, M. Lingelbach, R. Martín-Martín, F. Xia, K. E. Vainio, Z. Lian, C. Gokmen, S. Buch, K. Liu *et al.*, "BEHAVIOR: Benchmark for everyday household activities in virtual, interactive, and ecological environments," in *Conference on Robot Learning (CoRL)*, 2022. 2, 3

[74] S. Stepputtis, J. Campbell, M. Phielipp, S. Lee, C. Baral, and H. Ben Amor, "Language-conditioned imitation learning for robot manipulation tasks," *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. 3

[75] A. Szot, A. Clegg, E. Undersander, E. Wijmans, Y. Zhao, J. Turner, N. Maestre, M. Mukadam, D. S. Chaplot, O. Maksymets *et al.*, "Habitat 2.0: Training home assistants to rearrange their habitat," *Advances in Neural Information Processing Systems (NeurIPS)*, 2021. 2, 3

[76] J. Thomason, M. Murray, M. Cakmak, and L. Zettlemoyer, "Vision-and-dialog navigation," in *Conference on Robot Learning (CoRL)*, 2020. 2

[77] W.-C. Tseng, H.-J. Liao, L. Yen-Chen, and M. Sun, "CLA-NeRF: Category-level articulated neural radiance field," in *International Conference on Robotics and Automation (ICRA)*, 2022. 3

[78] W. Wan, H. Geng, Y. Liu, Z. Shan, Y. Yang, L. Yi, and H. Wang, "UniDexGrasp++ : Improving dexterous grasping policy learning via geometry-aware curriculum and iterative generalist-specialist learning," *arXiv preprint arXiv:2304.00464*, 2023. 2

[79] X. Wang, Q. Huang, A. Celikyilmaz, J. Gao, D. Shen, Y.-F. Wang, W. Y. Wang, and L. Zhang, "Reinforced cross-modal matching and self-supervised imitation learning for vision-language navigation," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2

[80] Z. Wang, Y. Chen, T. Liu, Y. Zhu, W. Liang, and S. Huang, "HUMANISE: Language-conditioned human motion generation in 3D scenes," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. 2

[81] F. Wei, R. Chabra, L. Ma, C. Lassner, M. Zollhoefer, S. Rusinkiewicz, C. Sweeney, R. Newcombe, and M. Slavcheva, "Self-supervised neural articulated shape and appearance models," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 3

[82] Y. Weng, H. Wang, Q. Zhou, Y. Qin, Y. Duan, Q. Fan, B. Chen, H. Su, and L. J. Guibas, "CAPTRA: Category-level pose tracking for rigid and articulated objects from point clouds," in *International Conference on Computer Vision (ICCV)*, 2021. 3

[83] F. Xia, W. B. Shen, C. Li, P. Kasimbeg, M. E. Tchapmi, A. Toshev, R. Martín-Martín, and S. Savarese, "Interactive Gibson benchmark: A benchmark for interactive navigation in cluttered environments," *IEEE Robotics and Automation Letters (RA-L)*, 2020. 2

[84] F. Xiang, Y. Qin, K. Mo, Y. Xia, H. Zhu, F. Liu, M. Liu, H. Jiang, Y. Yuan, H. Wang *et al.*, "SAPIEN: A simulated part-based interactive environment," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2, 4

[85] E. Xing, A. Gupta, S. Powers, and V. Dean, "KitchenShift: Evaluating zero-shot generalization of imitation-based policy learning under domain shifts," in *NeurIPS 2021 Workshop on Distribution Shifts: Connecting Methods and Applications*, 2021. 3

[86] Y. Xu, W. Wan, J. Zhang, H. Liu, Z. Shan, H. Shen, R. Wang, H. Geng, Y. Weng, J. Chen *et al.*, "UniDexGrasp: Universal robotic dexterous grasping via learning diverse proposal generation and goal-conditioned policy," *arXiv preprint arXiv:2303.00938*, 2023. 2

[87] A. Zeng, P. Florence, J. Tompson, S. Welker, J. Chien, M. Attarian, T. Armstrong, I. Krasin, D. Duong, V. Sindhwani *et al.*, "Transporter networks: Rearranging the visual world for robotic manipulation," in *Conference on Robot Learning (CoRL)*, 2021. 3, 6

[88] Y. Zhang and J. Chai, "Hierarchical task learning from language instructions with unified transformers and self-monitoring," in *Findings of the Association for Computational Linguistics (ACL-IJCNLP Findings)*, 2021. 2

[89] K. Zheng, X. Chen, O. C. Jenkins, and X. E. Wang, "VLMbench: A compositional benchmark for vision-and-language manipulation," *Advances in Neural Information Processing Systems Datasets and Benchmarks Track (NeurIPS Datasets and Benchmarks)*, 2022. 2, 3, 4, 6, 7